The visual compass: performance and limitations of an appearance-based method

Labrosse, Frédéric

Published in:
Journal of Field Robotics

DOI:
10.1002/rob.20159

Publication date:
2006

Citation for published version (APA):
Labrosse, F. (2006). The visual compass: performance and limitations of an appearance-based method. *Journal of Field Robotics*, *23*(10), 913-941. https://doi.org/10.1002/rob.20159

The visual compass: performance and limitations of an appearance-based method

Frédéric Labrosse

Department of Computer Science
University of Wales, Aberystwyth
Aberystwyth SY23 3DB, United Kingdom

e-mail: ffl@aber.ac.uk

This is a preprint of an article published in Journal of Field Robotics 23(10), 913–941 (2006), Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/rob.20159

The visual compass: performance and limitations of an appearance-based method

Frédéric Labrosse
Department of Computer Science
University of Wales, Aberystwyth, UK
ffl@aber.ac.uk

Abstract

In this article we present an algorithm to estimate the orientation of a robot relative to an orientation specified at the beginning of the process. This is done by computing the rotation of the robot between successive panoramic images, grabbed on the robot while it moves, using a sub-symbolic method to match the images. The context of the work is Simultaneous Localisation And Mapping (SLAM) in unstructured and unmodified environments. As such, very few assumptions are made about the environment and the robot's displacement.

The algorithm's performance depends on the value of a number of parameters being determined to provide overall good performance of the system. The performance is evaluated in different situations (trajectories and environments) with the same parameters and the results show that the method performs adequately for its intended use. In particular, the error is shown to be drifting slowly, in fact much slower than un-processed inertial sensors, thus only requiring infrequent re-alignment, for example when re-localising in a topological map. Limitations of the proposed methods are also shown and discussed.

1 Introduction

One of the difficult tasks a mobile robot faces when autonomously navigating is localisation, whether in a known or unknown environment (for example in the context of Simultaneous Localisation And Mapping, SLAM). Localisation is often limited to finding the position of the robot, assuming that the orientation is known from another source. For example, the orientation can be estimated using a known magnetic field, known landmarks such as the sun or stars, or by integrating rotational information provided by various sensors such as odometers, accelerometers or gyroscopes.

Using external, fixed information such as a magnetic field or stars is attractive since this does not require integration but provides an immediate estimate of the orientation. However, such data is not always available. For example, there is no reliable magnetic field on Mars

and Earth's magnetic field can vary enormously when close to other sources of magnetic fields (such as electrical equipment or ferrous ore bodies) or when close to the Earth's poles. Stars can also not be visible for example when indoors, underground, under an overcast sky or during day time.

Integrating rotational information has the major drawback of generally becoming less and less accurate as integration introduces additive errors at each step. In this article we describe a method to integrate rotational information to estimate the orientation of a robot. The rotational information is provided by a system that uses panoramic views of the environment of the robot grabbed by an omni-directional camera on the robot. The system only performs simple computations on the data, namely pixel shifts and distances between images. In particular, high-level (such as feature) extraction is avoided. Rather, only the appearance of the environment is used; the method is purely sub-symbolic. To this end, an algorithm will be described that uses these simple comparisons between successive images to offer a good compromise between robustness, reliability and portability as well as exhibiting low drift. Early results of this work have been presented elsewhere (Labrosse, 2004).

Section 2 introduces related work and describes where our work stands and can be used. Section 3 describes the theory behind the visual compass as well as practical improvements and parameters of the method. Based on these, an algorithm is devised and then evaluated with various datasets acquired in a range of situations covering both indoors and outdoors environments, Section 4. The method and results are discussed and future avenues of research are introduced, Section 5. A conclusion is presented in Section 6. Many results obtained during various experiments to develop and assess the visual compass are presented elsewhere (Labrosse, 2006).

2 Related work

Visual navigation increasingly relies on local methods: paths are specified in terms of intermediate targets that need to be reached in succession to perform the navigation task (Vassallo et al., 2002; Neal and Labrosse, 2004; Gourichon, 2004). This task can thus be reduced to a succession of homing steps (Binding and Labrosse, 2006).

Many of these homing methods that use vision require the orientation of the robot to be constant or at least known. This is for example the case of most methods derived from the *snapshot model* (Cartwright and Collett, 1983; Cartwright and Collett, 1987) ((Gourichon, 2004) gives a review and "genealogy" tree of such methods and (Ruchti, 2000) links biology and computational models). A snapshot is a representation of the environment at the homing position, often a one-dimensional black and white image of landmarks and gaps between landmarks, e.g. (Röfer, 1995; Möller et al., 1999), but also two-dimensional images of landmarks such as corners, e.g. (Vardy and Oppacher, 2003). Most of these methods use panoramic snapshots. Two notable exceptions to the orientation requirement estimate the change in orientation between the current and target orientations using balancing of the optical flow (Röfer, 1997) or by performing a search in a parameter space containing the change in orientation (Franz et al., 1997).

Vision-based localisation (Thompson et al., 1993) usually uses some transformation of the current view of the environment from the robot and matches it (often using a search in the pose space) with similar information synthesised from a map of the environment. For example, features of mountain images (such as peaks and ridges) can be extracted from range images (seen from the air) (Shaw and Barnes, 2003) or skyline (Naval Jr. et al., 1997; Cozman et al., 2000). In some cases, the orientation is determined at the same time as the position, which sometimes can lead to wrong matches (Neal and Labrosse, 2004), while in other cases, the orientation is determined using an external reference.

Such external reference can be obtained using a variety of sensors (and often a combination of these sensors). For example, a magnetic field can be used on Earth, while sun light polarisation can be used outdoors, both being cues used by insects, e.g. (Frier et al., 1996). The position of the sun can also be used (Cozman and Krotkov, 1995), or more generally alignment with landmarks or beacons. For example, methods using alignment with a remote landmark or linearly transforming the retinal position of three landmarks in panoramic images is proposed by (Gourichon, 2004). Note that this is a capability that insects seem to have, e.g. (Zeil et al., 1996; Graham et al., 2003). The Global Positioning System (GPS) can also provide orientation (or rather heading) information by measuring Doppler shift of satellite signal, sometimes combined with the derivative of motion information. Integrative methods can also be used. For example odometry measures wheel turn and infers the position and orientation of the robot with respect to an initial reference. This is extremely unreliable, especially in outdoors situations and/or with skid-steering (the type of robot used in the outdoors experiments reported here). A more reliable way is to use inertial sensors: accelerometers for position estimation and gyroscopes for angular estimation. These work by integrating the signal to provide the desired information. The main drawback of these integrative methods is accumulation of error. However, there is evidence that insects, in particular ants, use visual odometry and dead-reckoning to compute a homeward vector (Wehner et al., 1996; Srinivasan et al., 1997; Åkesson and Wehner, 2002) and such sensors have been used with success in robotic applications, e.g. (Hogg et al., 2002).

Using vision to compute the orientation is attractive, especially when it is also used to control the navigation. Indeed, this implies that less sensorial modalities are needed. Moreover, vision works everywhere (given the right camera) while other modalities often fail.

Almost all methods using vision for navigation extract features from the images and use these features to perform some matching. The features can be the landmarks of the snapshot and derived models, the skyline detected in (Cozman et al., 2000), etc. Recently, a large number of methods using the Harris detector or the Scale Invariant Feature Transform have been published, e.g. (Nistér et al., 2006; Lowe, 2004; Se et al., 2002). Moreover, landmarks can be characterised with a set of parameters such as size, area and contour length (Möller, 2001). However, extracting landmarks and possibly their parameters can be expensive and certainly implies assumptions about the world, in particular on its structure (Gonzales-Barbosa and Lacroix, 2002). Indeed, natural environments often present no obvious visual landmarks or when these exist, they are not necessarily easy to distinguish from their surroundings or sufficiently static. For example, a smoothly varying landscape on a foggy day will make the detection of features very difficult if not impossible. Moreover, matching (or recognising) landmarks can be difficult and therefore tends to be expensive, in particular in terms of neural

implementation (Möller et al., 1999). Indeed, most methods matching features between successive frames will assume that either the camera and/or features have not moved much, e.g. (Nistér et al., 2006), or that the camera underwent a constrained displacement, e.g. (Se et al., 2002). Without these assumptions, matching becomes prohibitively expensive and not robust enough.

Instead, we propose to use the images as they are, without pre-processing; this is the *appearance-based* approach (or signal-based approach in (Cozman et al., 2000)). In this approach, images are compared pixel by pixel to result in a single value measuring a quality of match between the compared images. Using whole two-dimensional images rather than a few landmarks extracted from images reduces aliasing problems; indeed, different places can look similar, especially if "seen" using only a few elements of their appearance. Moreover, such a method does not require any feature matching and will work even in the case of the smoothly varying foggy landscape mentioned above. Indeed, even a smooth gradient of colour can be correctly matched by the method proposed in this paper.

However, although the method does not explicitly use features, it does need features or, more loosely, variations in images. Typically, these will be distinctive patterns and coloured and/or textured patches. As mentioned above, variations in colour can be smooth and don't have to present high gradients. The pixel by pixel comparison will provide a good match when these "features" are aligned, thus providing a correspondence-less feature matching method. As will be discussed in Section 5, colour representation and pixel comparison are important aspects of appearance-based methods and are still largely unsolved problems. We propose in this paper a solution that works most of the time but that also has some well defined limitations.

Not many published papers propose to use raw images; a few examples follow. A one-dimensional panoramic image is used by (Röfer, 1995), from which the optic flow is extracted between successive images to control the robot. An array of light sensitive sensors (typically eight) is used by (Bisset et al., 2003) to represent and recognise places; a process similar to the one described in (Neal and Labrosse, 2004) is used to provide rotation independence. In (Gonzales-Barbosa and Lacroix, 2002), histograms of Gaussian derivative filtered images are used. The histograms constitute a somewhat reduced amount of information compared to using the whole images because the structure or spatial organisation of visual elements is lost. However, because the histograms are computed over rings of the images, some of the structure is preserved. Other publications mention different types of regions, e.g. (Gaspar et al., 2000; Jogan and Leonardis, 2000b). A detailed study of the pixel-wise comparison (Euclidean distance) between panoramic images captured in outdoors environments is done by (Zeil et al., 2003). Finally, (Franz et al., 1998) mention the possibility of using a method similar to the one we present in this paper to compute the change in orientation between current and target position but discards it because it becomes unreliable when the two compared images are too different. Although we do not necessarily agree with this, we propose here an incremental method that estimates the change in orientation by comparing successive images carefully chosen while the robot moves. The comparison is also designed for the task at hand and is a subset of more general comparisons between images performed in our algorithms for mapping (Neal and Labrosse, 2004) and homing (Mitchell and Labrosse, 2004; Binding and Labrosse, 2006).

The idea that motion information is present in the changes between successive images taken while the camera moves is obviously not new. In particular, panoramic images have been used in the past to extract information related to orientation of the camera, some having been mentioned above. Other methods, sometimes described as being appearance-based or correspondence-less, transform images into the frequency domain in which a translation becomes a change in phase. This is the case for example of (Pajdla and Hlaváč, 1999) and (Jogan and Leonardis, 2000b) for cylindrical images (the Zero Phase Representation, ZPR) and (Makadia and Daniilidis, 2006) for spherical images. However, there is very little evidence that these methods work for images taken from different positions. In fact, of the two brief experimental results in (Pajdla and Hlaváč, 1999), one shows that from images taken from slightly different positions but with the same orientation (no quantitative evaluation is given in the paper), a dramatic change in image orientation seems to be erroneously indicated by the shifted images.

Correlation of panoramic images transformed using the ZPR is used by (Jogan and Leonardis, 2000b). However, rotation invariance is obtained by (Jogan and Leonardis, 2000a) by condensing a subset of all possible orientations using an eigenspace of "spinning-images". Indeed, (Jogan and Leonardis, 2000a) argue that the ZPR is sensitive to noise, occlusions and changes in the camera's position, which makes the method not appropriate for our case.

3 The visual compass

We present in this section the visual compass, starting with the theory behind it and then proposing and discussing possible improvements. Parameters of the method are determined based on experiments and finally the algorithm is presented.

3.1 Appearance

In this work, the appearance of the environment surrounding a robot from a given pose is an image taken by the robot of the environment. The idea is not new and has been used in tasks such as recognition, e.g. (Bichsel and Pentland, 1994), and inspection, e.g. (Nayar et al., 1996). Directly using the appearance of the world as opposed to extracting features or the structure of the world is attractive because methods can be devised that do not need precise calibration steps as will be shown later.

Like others, e.g. (Jogan and Leonardis, 2000b; Cozman et al., 2000; Gaspar et al., 2000; Gonzales-Barbosa and Lacroix, 2002; Goedemé et al., 2005), we use panoramic images as they provide in one image everything that can be seen from the current position. More precisely, we use an omni-directional camera to grab omni-directional images, Figure 1[1]. The camera is made of a "normal" camera pointed upwards and looking at a hyperbolic mirror linked by a perspex tube to the camera, Figure 2. Front and back of the robot are respectively at the top and bottom of the omni-directional images. However, because of the configuration of the camera relative to the mirror used in our omni-directional camera, right

[1]The images coming from the camera (omni-directional and panoramic) have had their contrast greatly enhanced for better printing and had much lower contrast during the experiments.

Figure 1: An omni-directional image

Figure 2: The omni-directional camera: the "normal" camera (bottom) and the hyperbolic mirror (top) linked by a perspex tube

and left are inverted in omni-directional images.

For ease of processing, Section 3.3, omni-directional images are unwrapped into panoramic images that we define as the appearance of the world from the corresponding poses, Figure 3. The unwrapping excludes the parts of the omni-directional images that correspond to either the robot or the outside of the mirror and only keeps the part of the images corresponding to the surroundings of the robot.

The unwrapping is performed by scanning a line emanating from the centre of the omni-directional image and rotating it around the image by a fixed increment. Pixels of the panoramic image are taken along the line at regular intervals using the nearest pixel of the

Figure 3: The panoramic view corresponding to the omni-directional image in Figure 1: the appearance from the corresponding pose of the robot

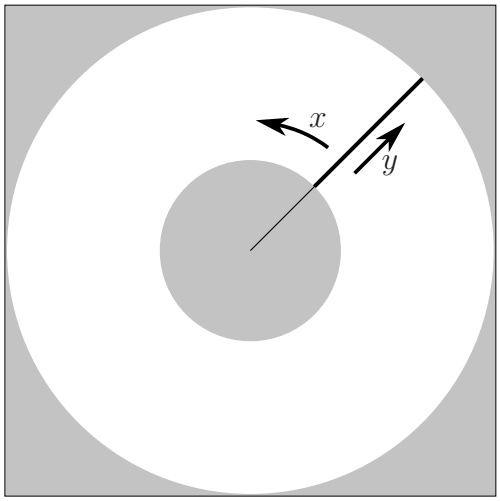

Figure 4: The unwrapping process. The grey areas correspond to unusable parts of the image (projection of the robot or outside of the mirror). x and y are the coordinates of pixels in the panoramic image.

omni-directional image[2], Figure 4. The direction of the unwrapping is such that the front and back of the robot are respectively at columns $0.25 \times w$ and $0.75 \times w$ from the left of the image, where w is the width of the panoramic image, Figure 3.

The unwrapping process is controlled by several parameters. The height of the panoramic images depends on the thickness of the white "doughnut" in Figure 4 and the sampling along the rotating line. This thickness is determined by the size of the omni-directional images and the size of the robot in the images. The sampling along the line can be non-linear to give more importance to various parts of the images. The outside of the doughnut mostly corresponds to the parts of the environment that are far away from the robot and/or that are tall while the inside corresponds to the parts that are close to the robot. Here we chose to use a linear sampling of one pixel in the panoramic image for each pixel along the line. The width of the panoramic images is also of importance. For ease of computation, it should be a multiple of 360, each pixel then representing a portion of a degree of angle of the environment surrounding the robot. This is further discussed in Section 3.4.

This unwrapping method does not produce a completely accurate representation of the environment, as the geometry of the mirror used in the omni-directional camera is not taken into account in the above transformation and no calibration of the system has been done.

[2]We have also used bi-linear interpolation in (Labrosse, 2004), but this is more expensive to compute and does not improve the performance of the system.

In particular, the perspex tube used to link the camera and the mirror is not perfect and introduces many local distortions. Moreover, the process assumes that the optical axis of the camera projects at the centre of the image, which is very probably wrong. However, as long as the distortions do not change dramatically with time, calibration is not needed as will be discussed later, Section 3.3. This means that the projection is not an accurate cylindrical projection.

3.2 Image space and appearance comparison

The input to the system is thus made of appearances (images) that need to be compared. An image of width w and height h with c colour components per pixel is a point in the *image space*, a space having $h \times w \times c$ dimensions representing all possible images of the given size.

Comparing appearances then becomes measuring the distance, using some distance metric, between points in the image space. The distance metric can be one of several. We have tried an L^1 norm (Manhattan distance) and an L^2 norm (Euclidean distance) in previous work (Mitchell and Labrosse, 2004) not showing any difference in the results obtained by either. Here we used the Euclidean distance. The distance between two images is thus defined as

$$d(\mathcal{I}_i, \mathcal{I}_j) = \sqrt{\sum_{k=1}^{h \times w} \sum_{l=1}^{c} (\mathcal{I}_j(k,l) - \mathcal{I}_i(k,l))^2}, \tag{1}$$

where $\mathcal{I}_i(k,l)$ and $\mathcal{I}_j(k,l)$ are the l^{th} colour component of the k^{th} pixel of images $\mathcal{I}_i$ and $\mathcal{I}_j$ respectively. Pixels are enumerated, without loss of generality, in scan-line order from top-left corner to bottom-right corner. In all experiments presented here, we used the RGB (Red Green Blue) colour space, thus having three components per pixel.

The choice of the Euclidean distance is driven only by its simplicity and it not introducing any discontinuities. Note that the square root in Equation (1), although not needed, has been kept not to give too much importance to large distances. Similarly, using the RGB colour space is not optimum because different colours will contribute differently to the comparison only because their RGB encoding is different, not because they are more prominent. These issues will be discussed further in Section 5.

Moreover, RGB is dependant on changes in illumination and other colour spaces would be better in that respect (Woodland and Labrosse, 2005). Again, however, since we only compare successive images, chances are that illumination changes will not be dramatic. Moreover, RGB being the space used by the frame grabber, using it means that no further computation is needed. Note that colour is probably not needed in this work and only using grey images would speedup the algorithm, although the RGB images provided by the frame grabber would have to be converted. No formal comparison has been made. However, colour images do provide more information than grey images and this is important to help solving aliasing problems in mapping and localisation. Given that the visual compass is a component of our work in this area, colour has been kept.

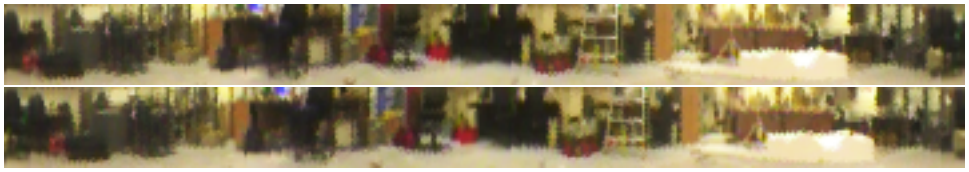

Figure 5: Two appearances from the same position but with a rotation of 10°

3.3 Extraction of rotational information

Using the appearance of the robot's environment is only useful if one can extract information from it about the environment that is useful to the robot. In this work, we are interested in rotational information.

Note that translational information is also present in changes of the appearance when the robot translates. This has been experimented with by (Zeil et al., 2003) and used for example by (Binding and Labrosse, 2006).

3.3.1 The perfect case

We assume for now that the system is perfect. This would imply several constraining, even impossible, assumptions, namely that:

- the optical axis of the camera is aligned with the axis of the mirror,
- the camera assembly (including the mirror and perspex tube) is calibrated,
- the axis of the system is aligned with the axis of rotation of the robot and the robot **only** turns on the spot (i.e. there is no translation),
- the resolution of the images is infinite.

These will be relaxed later! Rotating the robot on the spot would then result in a simple column-wise shift of the appearance in the opposite direction. The exact rotation angle could be retrieved by simply finding the best match between the first image (before rotation) and a column-wise (with column wrapping) shift of the second image (after rotation). The best shift would simply correspond to the rotation undertaken by the robot. Figure 5 shows two appearances different only in an almost perfect rotation by 10° of the robot between the two appearances. The rotation was obtained by rotating the first omni-directional image into the second omni-directional image, both images being then unwrapped to produce the appearances. This does not completely satisfy the assumptions because the centre of the omni-directional images is only approximately the projection of the axis of rotation of the optical system and because the resolution of the images is not infinite. Figure 6 show the Euclidean distance between the two images as a function of the column-wise shift (rotation) of the second image:

$$d(\mathcal{I}_i, \mathcal{I}_j, \alpha) = \sqrt{\sum_{k=1}^{h \times w} \sum_{l=1}^{c} (\mathcal{I}_j(\alpha, k, l) - \mathcal{I}_i(k, l))^2}, \tag{2}$$

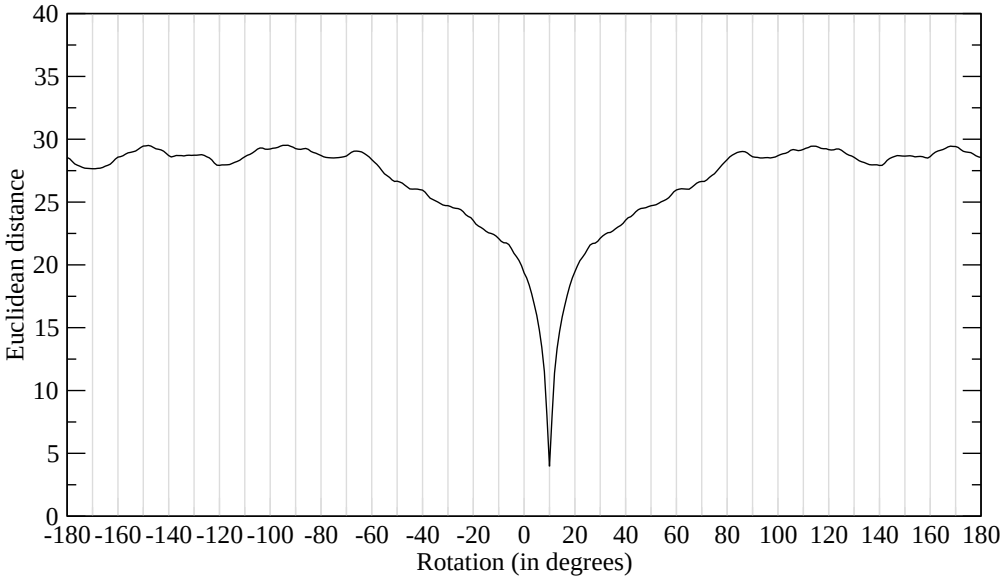

Figure 6: The Euclidean distance between the two images in Figure 5 as a function of the column-wise shift of the second image

where $\mathcal{I}_i(k,l)$ and $\mathcal{I}_j(\alpha,k,l)$ are the l^{th} colour component of the k^{th} pixel of images $\mathcal{I}_i$ and $\mathcal{I}_j$ respectively, the latter being column-wise shifted (with column wrapping) by α pixels, corresponding to α degrees because the images are 360 pixels wide. The minimum of the function is clearly obtained at 10°. Note that for ease of visualisation, the rotations are shown on graphs in the interval $[-180;+180]$ but are actually internally used in the interval $[0;360)$ to match the output of a "traditional" compass. Similarly, all arithmetic on rotations and column-shifts is done modulo 360 and w respectively.

Finding the rotation between the two images is thus finding the best shift α_m that will minimise the distance:

$$d(\mathcal{I}_i,\mathcal{I}_j,\alpha_m) \leq d(\mathcal{I}_i,\mathcal{I}_j,\alpha) \quad \forall\alpha \in \mathbb{R}. \tag{3}$$

The rotation between the two images is:

$$\rho = -\alpha_m \times \frac{360}{w}. \tag{4}$$

Shifting column-wise the second image by $-\alpha_m$ columns to make it match as well as possible the first image is what we call the *un-rotation procedure*. Note that this procedure does not have to be actually performed: this is simulated using pointer arithmetic during the computations described in this paper.

The minimum distance in Figure 6 is not 0 as both the manually performed rotation of the omni-directional image and the re-sampling done during the unwrapping process introduce errors.

It can be seen that the function presents many local minima. However, if one assumes that the rotation between the two appearances is not too large, then finding a local minimum is enough when one starts from 0. Alternatively, if an estimation of the change in orienta-

tion is available, again a local minimisation is enough. These aspects will be discussed in Section 3.3.4.

Finding the orientation of the robot relative to a starting given or known orientation is done by adding successive changes in orientation between successive frames using Equations (3) and (4).

3.3.2 The finite resolution case

Obviously, we do not have an infinite angular resolution. This is because omni-directional images have a finite resolution (a hardware constraint!) and the unwrapping procedure cannot extract more information than that present in the original images. This means that a quantisation error is made when evaluating the rotation using only column-wise shifts. However, the error is at most $\pm 0.5^\circ$ for each pair of images if the angular resolution is 1 pixel per degree. Ways of improving this will be discussed later.

In this case, only a discrete approximation of the continuous rotation can be obtained. Equations (3) and (4) respectively become:

$$d(\mathcal{I}_i, \mathcal{I}_j, a_m) \leq d(\mathcal{I}_i, \mathcal{I}_j, \alpha) \quad \forall \alpha \in \mathbb{N}, \tag{5}$$

and

$$r = -a_m \times \frac{360}{w}. \tag{6}$$

3.3.3 The general case

One of the advantages of appearance-based methods compared to full reconstruction is that they do not need any accurate calibration or even perfect optics.

The assumptions mentioned in the perfect case above are rather unrealistic and at least constraining. For example, such optical systems as the omni-directional camera can be physically difficult to implement and to ensure proper reliable alignment would require it to be prohibitively heavy[3]. Moreover, calibration of the system would have to be performed regularly as such a system might be deformed/damaged in situations such as planetary exploration or during landing on remote places. Moreover, the robot usually does not rotate on the spot if only because it usually also translates while turning.

All these imperfections mean that a rotation of the robot will result in more than a simple column-wise shift of pixels in the appearance. Indeed, the viewpoint will change introducing new features in the appearance while others will disappear. However, provided the change in view point is not too dramatic, the method can still be used. For example, Figure 7 shows the appearances before and after a displacement of 20 cm followed by a rotation of -30°. Figure 8 shows the Euclidean distance between the two appearances as a function of the column-wise shift of the second appearance. The minimum of the function is at -30°, indicating the correct rotation of the robot, despite the change in position resulting in a

[3]This is not completely true anymore since compact omni-directional cameras can now be bought off-the-shelf, although their suitability for the present task hasn't been evaluated by the author.

Figure 7: Two appearances before and after a displacement of 20 cm and rotation of $-30°$

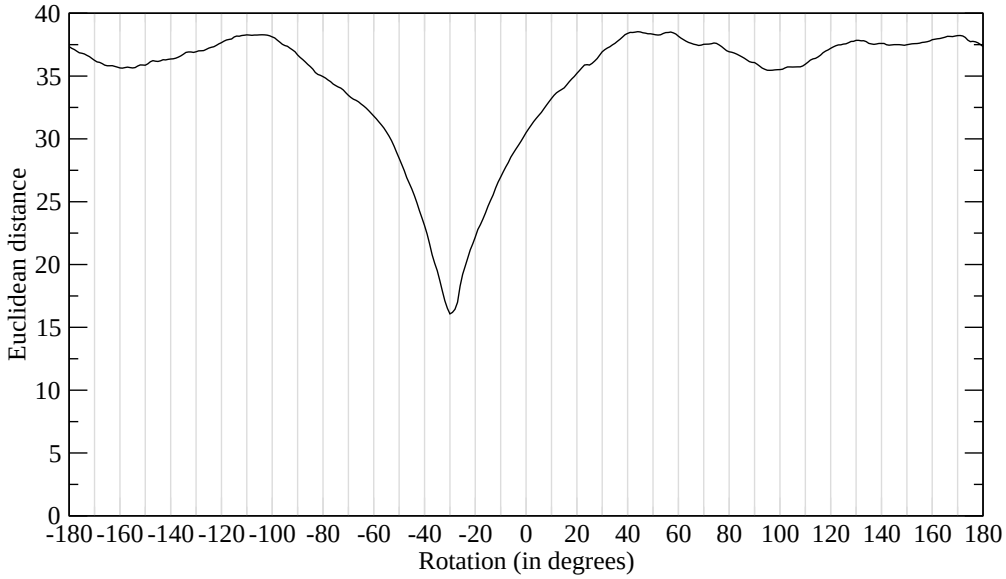

Figure 8: The Euclidean distance between the two images in Figure 7 as a function of the column-wise shift of the second image: the general case

non-exact match, which is visible in the higher values of the distance function, compared to the perfect case.

Dramatic changes in view point are created either by large displacements or by small displacements while the robot is close to (large) objects. Indeed, important changes in the appearance happen at occlusion points in the images. However, since we do not seek an exact match but only the best match, the method still works, even when the constraints in Section 3.3.1 are not exactly satisfied. The only constraint is that changes in appearance due to occlusions are small and distances in Cartesian space between the robot and obstacles are larger than the robot's displacement between successive images. This constraint is usually satisfied since robots tend to either navigate away from obstacles or, when near obstacles, move slowly. Pitch and roll of the robot will also introduce changes between consecutive images that are not recoverable. However, provided that the world contains tall features that will overlap in successive images in case of pitch and roll, the matching will still work.

3.3.4 Minimisation of the distance function

To obtain the change in orientation, a minimisation must thus be performed for each pair of considered images, Equation (5). For this, we perform a local minimisation that, if

the rotation between tested images is not too important, will lead to the global minimum. Indeed, as can be seen, e.g. in Figure 8, the distance function presents a clear minimum that is attainable if the minimisation starts from a good estimate, which can be 0 if the rotation is not too important. However, as will be described later, the rotation might not be small enough and/or the distance function might not behave as well as needed, e.g. Figure 21, resulting in failures of the local approach. To solve this, we perform a combination of a local and a global search: the local minimum is first sought using a simple descent algorithm, which in most cases is successful and does not need more than $r + 1$ evaluations of the distance, where r is the rotation between the two frames. From this local minimum, a number of positions on both sides of it are evaluated in turn. If any of them is lower than the previous minimum, then the local minimisation is started again from that position. In all experiments reported here, eight values spanning a rotation of 20° centred on the local minimum were evaluated. If any of these values is indeed lower, then the previous local minimisation only had to evaluate a small number of values given the shape of the function, if the initial starting position was not too wrong. The algorithm described in Section 3.5 ensures that this is the case. This minimisation thus usually does not need a large number of evaluations of the distance function (generally no more than the number of degrees of the rotation between successive images).

3.3.5 Limitations

Before even running experiments, some limitations of the method described above can be expected. However, these are rather pathological and similar to human (or at least biological) limitations and are inherent to systems using vision.

The method needs visible features in the environment, despite not explicitly using them. Indeed, if the environment is completely featureless, then the distance in image space will remain constant (and null) and the method will fail. However, even low contrast or smoothly varying colours in the appearance is enough to measure differences in images, while methods based on explicit feature extraction will usually fail with such images.

Checkerboard-type environments might also make the method fail, depending on the spatial sampling rate used: if the rate is too low, aliasing might occur, i.e. different places will produce similar (if not equal) images and the rotation computation will fail. The aliasing problem however will not disappear if the rate is made higher, but the rotation computation will succeed.

A more subtle problem is the "one-feature-syndrome". If the robot is moving in a mostly homogeneous environment containing only one prominent visual feature, such as a single tree in the desert, then the feature will move on the images. However, because only the pixels corresponding to the feature will be different, the un-rotation procedure will make that feature match at the same position in the succession of images, resulting in the system reporting a rotation even when none occurred or on the contrary no rotation when one did occur, depending on where the feature is — the unique, or dominating, feature "pulls" (or "pushes" in some cases) the stabilisation process. This however, is exactly what happens to human beings in similar situations, e.g. on board a boat at sea in the night when passing

another boat. We thus have the constraint of having at least two distinguishable features in the environment that should be isotropically arranged (as is the case for most methods derived from the *snapshot model*), i.e. homogeneously spread in space around the robot. However, because we use the appearance, as opposed to landmarks, of the world, this usually is the case. The *equal distance assumption* (Franz et al., 1998) states that the landmarks must be at the same distance from the position of the snapshot. This again is also an assumption made here because different distances create different amplitudes of the optic flow, leading to different apparent rotations. However, because we use only a global measure, this is not such a problem here. These two assumptions will be revised later when the field of view will be reduced, Section 3.4.3. Some of the results presented in Section 4.3 provide examples of this effect, which is discussed further in Section 5.

3.4 Practical improvements

The conditions imposed by the theory not being met, a number of practical improvements have been explored to improve the performance of the system. For this, experiments have been conducted on a number of image sequences grabbed in various situations, some of which are presented in Section 4.2. These experiments are detailed in (Labrosse, 2006) and only the results are presented here. The experimental setup of these experiments is described in Section 4.1.

Early experiments showed that the orientation, when computed using Equations (5) and (6) drifts dramatically. In particular, if the rotation of the robot is constant, the drift is linear with the image index. This is because each measure of the rotation between consecutive images introduces a small error that accumulates as we proceed. If the rotation between successive frames is constant, or more generally the conditions are wrong (see Section 4.3 for some cases), then the errors never quite cancel each other, resulting in the drift. A large number of experiments to describe this is reported in (Labrosse, 2006).

Several solutions to that problem have been explored: higher angular resolution of the panoramic images, interpolation of the distance measurement, reduction of the field of view and better spatial sampling. We will look at each of these in turn.

In each case, experiments were performed (reported in (Labrosse, 2006)) and the performance of the system was evaluated using the error between computed and measured orientations at the end of sequences of images in various situations (Section 4 gives more details).

3.4.1 Angular resolution

Part of the error is due to the discrete nature of the data used. The resolution of the measurements is limited by that of the panoramic images, which in turn is limited by that of the omni-directional images. For example, if the omni-directional image is 400 pixels wide (and high), then the highest angular resolution that can be attained is 3.49 pixels per degree. Indeed, a circle of diameter 400 pixels provides $1,256$ pixels on its circumference, thus at the top row of the appearance image. The other rows will still have the same number of columns; their angular resolution will not be as high and will thus contain redundant

information. Note that most of the pixels from the outer parts of the omni-directional images are preserved thus using as much as possible of the already low spatial resolution of that part of the images.

Results on all datasets show that the performance dramatically improves (from errors of about 175° to errors below 5° at the end of a particular image sequence) when the angular resolution is increased up to the maximum resolution available in the panoramic images (up to three pixels per degree), at which point the error presents only a very low drift (less than 5° for a total rotation of about 1080°). However, increasing the angular resolution is expensive at all stages of the process (grabbing, unwrapping and distance measurement) and this improvement has therefore not been used.

3.4.2 Interpolation of the distance function

Increasing the angular resolution can be advantageously replaced by interpolating the distance between images as a function of rotation of the second image (Equation (2)) around the pixel-resolution minimum.

If one is only interested in the position of the minimum, then a simple parabolic interpolation using the pixel-resolution minimum and the two points on either side of it proves to be sufficient. This can be tested by comparing the sub-pixel minimum with that obtained using higher angular resolution images. No formal evaluation has been conducted by the author but visual inspection of many cases and the better results in orientation estimation prove that the method is viable.

However, if one is also interested in obtaining a better minimum (the height of the function), as will be the case, then a better scheme must be used. Indeed, the parabolic interpolation provides a good estimate of the value of the minimum only when the two compared images are sufficiently distant in image space after best match un-rotation: in this case, the distance function is smooth enough for that, see for example Figure 8. However, when the two images are similar, then the minimum is sharp and the parabolic interpolation is not anymore a good model, Figure 6. A better model is obtained by using linear extrapolation of the two slopes on either side of the sub-pixel minimum. This can be determined by comparing the three lowest values of the distance function d_m, d_1 and d_2 respectivally corresponding to rotations a_m, a_1 and a_2 (with $d_m \leq d_1 \leq d_2$). Linear extrapolation is more suitable when we have $d_1 < (d_2 + d_m)/2$, Figure 9 (left), while parabolic interpolation is more suitable otherwise, Figure 9 (right). The estimation of the sub-pixel minimum using linear extrapolation is $\hat{\alpha}_m$ while the one obtained using parabolic interpolation is $\tilde{\alpha}_m$. The corresponding values of the distance function are respectively $\hat{d}_m$ and $\tilde{d}_m$.

As Figure 12 shows, the distance function progressively changes from a sharp minimum for similar images to a smooth minimum for different images. We thus model the sub-pixel minimum by smoothly going from linear extrapolation to parabolic interpolation. For this, we need to define the amplitude of the distance function between two images as:

$$\mathcal{A}(\mathcal{I}_i, \mathcal{I}_j) = d(\mathcal{I}_i, \mathcal{I}_j, \alpha_m + w/2) - d(\mathcal{I}_i, \mathcal{I}_j, \alpha_m). \tag{7}$$

This is the difference between the distances for the worst match and the best match. In fact,

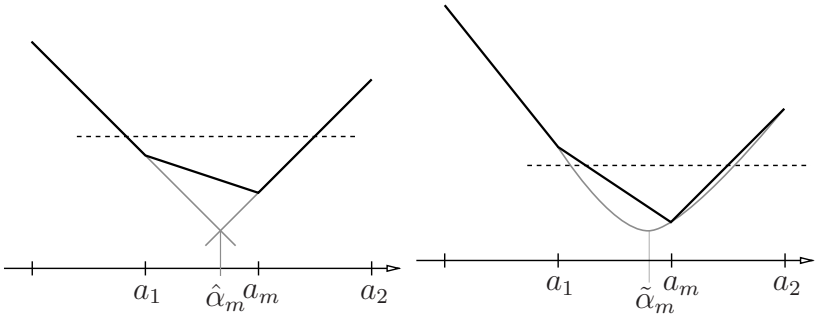

Figure 9: The two possible configurations of the minimum of the distance function. Symmetric configurations are created by swapping a_1 and a_2. The bold black line shows the pixel-resolution distance function around the minimum while the thin grey lines show the linear extrapolation (left) and parabolic interpolation (right).

the maximum of the distance function is usually not at $\alpha_m + w/2$ but this corresponds to the rotation opposite the best rotation and is stable in value as can be seen on Figure 11. Because the amplitude depends on the colour content of the images, we normalise the amplitude with that of the amplitude of the distance function of the first image and itself:

$$\mathcal{A}_n(\mathcal{I}_i, \mathcal{I}_j) = \frac{\mathcal{A}_(\mathcal{I}_i, \mathcal{I}_j)}{\mathcal{A}_(\mathcal{I}_i, \mathcal{I}_i)}. \tag{8}$$

To provide a monotonically decreasing amplitude as a function of the distance in image space between the two images, the amplitude was made, in Equation (7), dependent on the sub-pixel values of the distance function, which are not yet available. We also define an amplitude using the pixel-resolution minimum:

$$A(\mathcal{I}_i, \mathcal{I}_j) = d(\mathcal{I}_i, \mathcal{I}_j, a_m + w/2) - d(\mathcal{I}_i, \mathcal{I}_j, a_m), \tag{9}$$

and similarly for the normalised amplitude A_n.

A careful study of several graphs similar to the one in Figure 11 tells us that for normalised amplitudes below 0.5 the parabolic interpolation is sufficient. However, for normalised amplitudes close to 1.0, the linear extrapolation is better. We therefore linearly interpolate between the values obtained by linear extrapolation and parabolic interpolation to compute the estimation of the sub-pixel minimum of the distance function:

$$d_m(\mathcal{I}_i, \mathcal{I}_j) = \begin{cases} \tilde{d}_m & \text{if } A_n(\mathcal{I}_i, \mathcal{I}_j) \leq 0.5 \\ p \times \hat{d}_m + (1-p) \times \tilde{d}_m & \text{otherwise} \end{cases}, \tag{10}$$

where $p = (A_n(\mathcal{I}_i, \mathcal{I}_j) - 0.5)/0.5$. The sub-pixel shift is estimated similarly:

$$\alpha_m(\mathcal{I}_i, \mathcal{I}_j) = \begin{cases} \tilde{\alpha}_m & \text{if } A_n(\mathcal{I}_i, \mathcal{I}_j) \leq 0.5 \\ p \times \hat{\alpha}_m + (1-p) \times \tilde{\alpha}_m & \text{otherwise} \end{cases}. \tag{11}$$

The sub-pixel amplitude can now be computed as:

$$\mathcal{A}(\mathcal{I}_i, \mathcal{I}_j) = d(\mathcal{I}_i, \mathcal{I}_j, \alpha_m + w/2) - d_m(\mathcal{I}_i, \mathcal{I}_j). \tag{12}$$

The minimum thus computed ensures in most cases that increasing the distance in image space between images gives a monotonically increasing minimum, Figures 11 and 12 and therefore monotonically decreasing normalised amplitude.

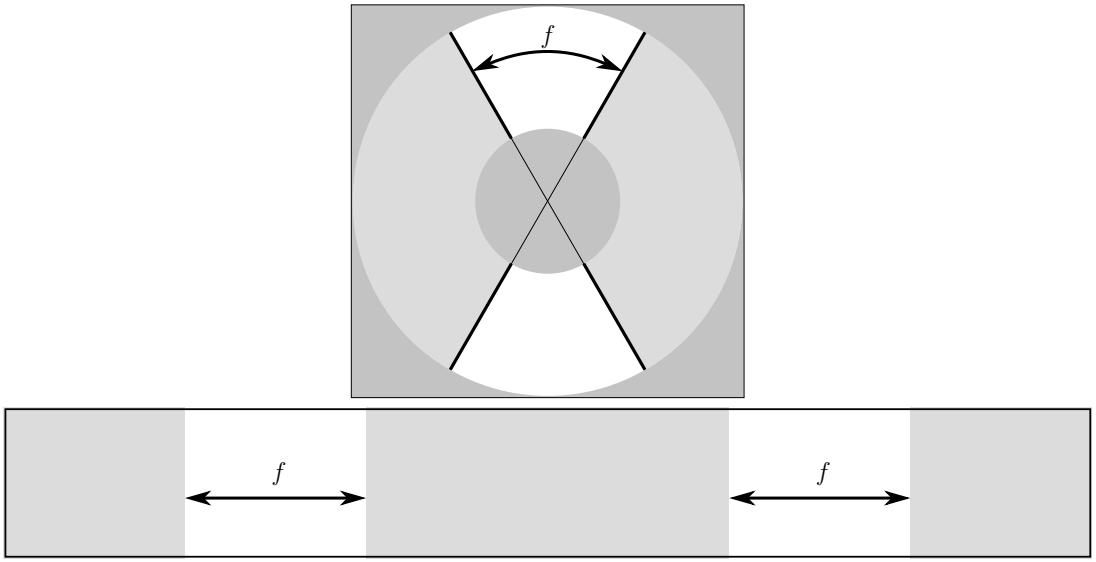

Figure 10: The reduced field of view in white in the omni-directional (top) and panoramic (bottom) images

Note that most of the distance values needed for the interpolation are computed anyway during the minimisation. Moreover, because of the configuration of the points involved, the arithmetic can be simplified to the extreme. This is therefore not an expensive procedure.

The improvement of the method due to interpolation is not as dramatic as the one provided by an increase in angular resolution. However, the interpolation is computationally not expensive and has thus been preferred over the increase in angular resolution and is used in the results presented from now on, unless otherwise stated.

3.4.3 Field of view

Another improvement that can be done over the theoretical method is to only consider parts of the images. Indeed, as noted by (Nelson and Aloimonos, 1988), the contribution to the optic flow by the motion of the robot is not homogeneous in omni-directional images; a forward/backward translation mostly contributes in the regions corresponding to the sides of the robot and very little in the parts corresponding to the front and back of the robot while the rotation contributes equally everywhere. Because we are interested in extracting the rotation information, only considering the regions of the images corresponding to the front and back of the robot allows us to discard most of the problems introduced by the translation, in particular sudden changes in appearance (parallax). The reduced field of view is shown in Figure 10.

In our case, the robot cannot undergo sideways translations and the front and back are therefore always positioned at the same place in the panoramic images. However, some robots can translate sideways. In such a case, some estimation of the direction of the translation is needed to obtain the reduced field of view. Although we have not tested this, it is expected that the performance of the system is not particularly dependent on the exact position of

the front and back of the robot.

To use the reduced field of view, the images are actually not modified in anyway. Only the pixels in the white areas of Figure 10 are used in the matching process. When column-wise shifting the second image in Equation (2), the reduced field of view is also shifted to align it with the current estimation of where the front and back are in the second image. Because all the pixels are always available, there is no boundary effect to consider.

Experiments, reported in (Labrosse, 2006), show that keeping the areas of width $f = 60°$ around the front and back of the robot, Figure 10, provides a good compromise between losing too much of the environment and allowing too much parallax error. This also provides an important improvement over using the whole field of view in terms of stability and sensitivity to prominent features at the sides of the robot. This is used from now on, unless otherwise stated.

3.4.4 Spatial sampling

Because of un-avoidable errors made at each estimation of the rotation due to limited resolution of the images, it could be a good idea to use as few images as possible (in other words, to sample as seldom as possible the Cartesian space). However, the more distant (in Cartesian and image space) the images are, the more error is made in the computation because of parallax. Therefore, there is a trade-off between accumulating small errors often and accumulating large errors but less often. Experiments reported in (Labrosse, 2006) show that lowering within limits the frame rate does indeed improve the error from about 175° to a few degrees on the same dataset mentioned in Section 3.4.1, without the previously mentioned improvements.

The error is due to both a non perfect system and parallax due to changes in viewpoint while the robot moves. These have been addressed to some extent in the previous sections. However, a better analysis can be done of the parallax error and its causes. The amount of change in viewpoint and thus parallax error depend on the speed of the robot, the capture rate and more importantly the structure of the environment, which we do not know. However, the distance function between pairs of images (Equation (2)) is itself a function of the distance between the two images in image space and Cartesian space. Indeed, the shape of the function as well as the values the function takes vary smoothly as the image space distance between the two images changes. Figures 11 and 12 show the distance between a reference image and successive following images as a function of the column-wise shift of the second image of the pairs for images taken from the dataset CIRCLE1, Section 4.2. It is clear that the top parts of the curves are all similar (in fact differ mostly by a horizontal shift due to the rotation of the robot between the different frames). The bottom part of the curves changes smoothly in two ways: a horizontal shift shows the rotation of the robot while an increase of the minimum shows increasing parallax error between frames. It is thus clear that this minimum is a good indication of evolution of the difference between images. The method to automate the space sampling rate is thus to set a threshold on the normalised amplitude of the distance between a reference image and the current image (Equation (8)) and to skip images until the threshold is exceeded (normalised amplitude becomes lower

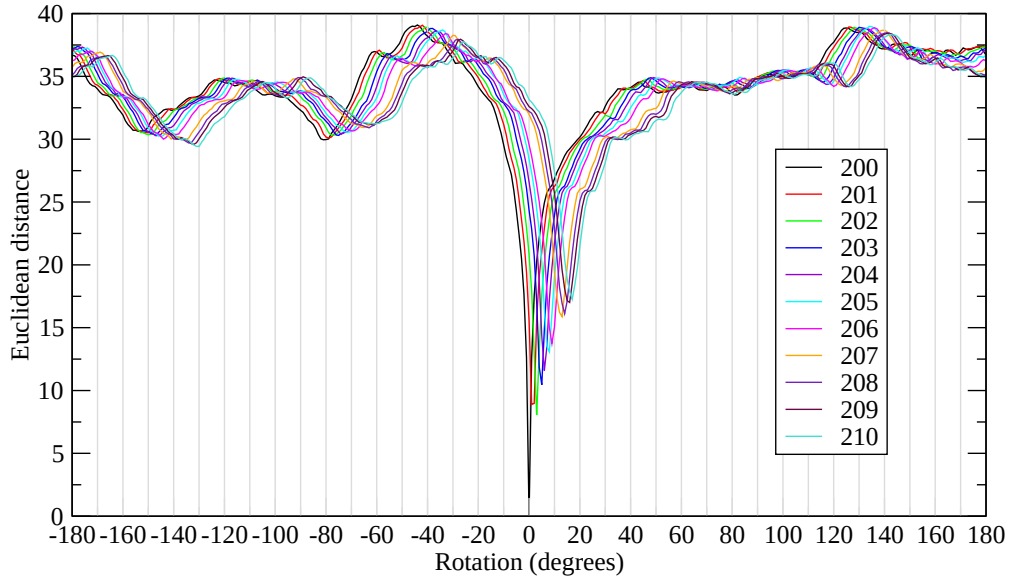

Figure 11: CIRCLE1: the Euclidean distance between image 200 and images 200 to 210 as a function of the column-wise shift of the second image of the pairs

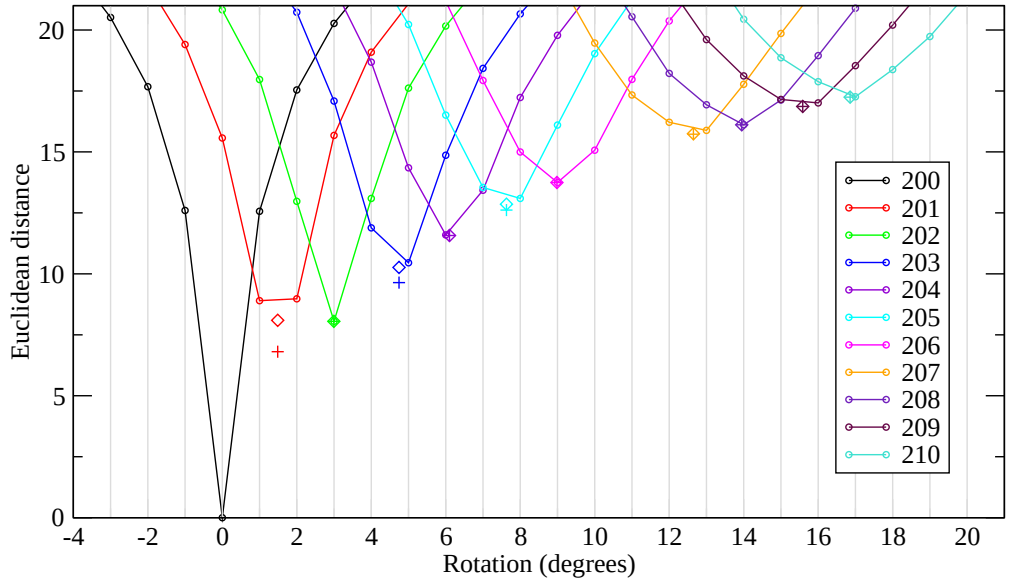

Figure 12: CIRCLE1: close-up of Figure 11. The '◇' show the interpolation using parabolic interpolation while the '+' show the mixture of linear and parabolic interpolation.

than the threshold).

To set the threshold, we manually determined what the best spatial sampling rate was for a given dataset (CIRCLE1) by skipping various numbers of frames at a time. This was done using interpolation and the reduced field of view as described above. The best result was obtained for a regular sampling rate corresponding to skipping three frames at a time, i.e. using one frame in four. For this best regular sampling rate, we measured the normalised

Table 1: CIRCLE1: statistics on the normalised amplitude of the distance for the best value of the skip (3) and the following

Skip	Mean	Std. Dev.	Median	Min.	Max.
3	0.6413	0.0486	0.6430	0.4881	0.7617
4	0.6053	0.0541	0.6055	0.3448	0.7284

amplitude of the distance function (Equation (8)) between all pairs of kept images and calculated some statistics on the obtained values, which are given in Table 1. The relatively low standard deviation as well as the fact that means and medians are close show that the distribution is compact. On the grounds that we want to sample rather less often than more, the threshold was taken as the median value corresponding to the best skip plus one, any value between the two limits being acceptable. The median value for a skip of four is thus used as the threshold on the normalised amplitude in the algorithm described next.

As the results will show, Section 4.3, the performance of the system when using the threshold is slightly worse than when using a constant, adequate, manually determined sampling rate. However, the threshold offers a method that works well (but not optimally) in all cases while the constant rate is very dataset dependent.

3.5 The algorithm

The method is summarised in Algorithm 1. In the algorithm, it is assumed that the panoramic images have one pixel per degree (as was the case in all experiments). If this is not the case, then the rotation must be computed (Lines 9 and 12) from the shift using Equation (4).

```
1:  t ← 0.6055        // Threshold on normalised amplitude
2:  I_r ← I_c ← newImage()        // Reference and current images
3:  ω_r ← ω_c ← 0        // Reference and current orientations
4:  r_rc ← 0        // Rotation between reference and current image
5:  loop
6:    I_p ← I_c        // Previous image
7:    I_c ← newImage()
8:    r_rp ← r_rc        // Rotation between reference and previous image
9:    r_rc ← −α_m(I_r, I_c)
10:   if A_n(I_r, I_c) < t then        // Reference and current images are too different
11:     ω_r ← ω_r + r_rp
12:     r_rc ← −α_m(I_p, I_c)
13:     I_r ← I_p
14:   end if
15:   ω_c ← ω_r + r_rc
16: end loop
```

Algorithm 1: Estimation of the orientation

Table 2: The parameters of the visual compass algorithm

Angular resolution of the appearance	1 pixel per degree
Size of the appearances	360×45 pixels
Distance interpolation	mixture of parabolic interpolation and linear extrapolation
Front and back field of view	60°
Distance minimisation	local + regular scanning
Space sampling rate	threshold on the normalised amplitude: 0.6055

The orientations can be initialised with any initial value (0 in the algorithm, Line 3). At all times, ω_c contains the current estimation of the orientation from the orientation at the beginning of the process.

On Line 12 of the algorithm, the rotation between reference and current images is recomputed using the previous and current images because the success of the test (Line 10) means that the current image is too different from the reference image and therefore the estimation of the rotation might be wrong.

Line 11 provides a firewall as in (Nistér et al., 2006): the orientation of the reference image is only updated when the current image becomes too different.

In the actual implementation, the minimisation is normally performed only once (Line 9) and is started from r_{rp}, which is close to the expected value of r_{rc} thus allowing a fast local minimisation, Section 3.3.4. However, when the reference image is changed, an additional minimisation is performed (Line 12) between the previous and current images (two consecutive images) starting from a rotation of 0, which is again close to the expected rotation.

4 Experiments

In this section, we apply the algorithm with the parameters estimated and evaluated in previous sections to different datasets to evaluate the performance of the visual compass. The parameters are recapitulated in Table 2.

4.1 Experimental setup

The experimental setup is as follows. The omni-directional camera, Figure 2, is mounted on a mobile robot, either a Pioneer 2DXe for the indoors experiments, Figure 13, or a Pioneer 2AT for the outdoors experiments, Figure 14. The camera is mounted such that its axis is as close as possible to the axis of rotation of the robot. However, in the spirit of appearance-based methods, no proper calibration of this arrangement has been made. The outdoors robot uses skid-steering to turn, thus producing non-smooth rotations, especially on high-grip surfaces such as the car park used later.

Images are grabbed and can be processed on the robot. However, in all experiments reported

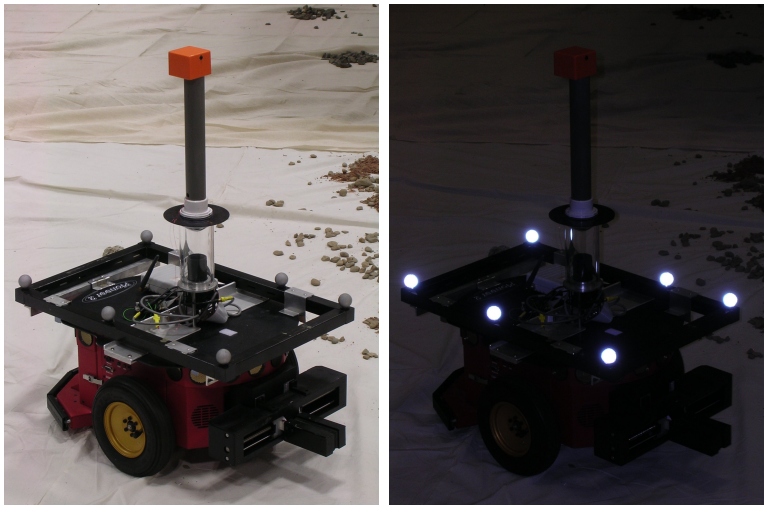

Figure 13: The indoors robot with the camera, the magnetic compass and the reflective markers

Figure 14: The outdoors robot with the camera and magnetic compass

here, the images are saved when grabbed and batch-processed later to be able to perform different experiments on the same data set with different values for the different parameters, thus providing comparable results.

The box at the top of the mast above the camera, Figures 13 and 14, contains a magnetic compass used to assess the visual compass during outdoors experiments. The compass is mounted high enough to avoid magnetic interference created by the robot. However, the compass is disturbed by the presence of metal in the floor of our lab as well as all electronic equipment in the lab and therefore has not been used indoors. Outdoors, the compass is also sensitive, to a lesser extent, to pitch and roll of the platform. The performance of the magnetic compass will be characterised during outdoors experiments.

Indoors, the performance of the visual compass is evaluated using the real-time motion tracking system VICON 512. The system tracks and provides in real-time the position of

reflective markers. Objects can be defined in the system as a set of markers shown to it. The objects can then be tracked in real-time (at between 70 and 120 frames per second on the computer used for the experiments reported here) and a software server can then provide real-time data about the position and orientation of the object to any software client connected to it. Figure 13 shows the frame on the indoors robot along with the reflective markers.

The performance of the VICON system is difficult to asses without the help of another measuring device (such as a theodolite). The VICON documentation states errors of the order of the millimetre on the position of the markers, which leads to an angular error of about $0.35°$ for the tracked markers of the experiments. However, for our experiments, the important properties of the orientation as returned by the VICON system (or the magnetic compass) is that it is absolute and thus not drifting, a property that needs to be evaluated for the proposed system. Moreover, the error in the orientation provided by the VICON system is likely to be at least one order of magnitude lower than that of the visual system.

To allow comparison, when an absolute orientation is available, either from the VICON system or the magnetic compass, the visual compass is initialised with the absolute value at the beginning of the experiment.

The two robots are identical software-wise and use GNU/Linux as their operating system on a PC104 format Pentium III at 800 MHz. The robots are connected to the Internet via a slow wireless connection through a firewall protecting the main wired network from the wireless network. The VICON server is on the wired network side. The robots must thus get the VICON data from the network. The software implementing the visual compass and grabbing all the necessary data for its evaluation runs, on the robot, a tight loop without threading to provide maximum control over the timing of its execution and in particular the synchronisation between the different sources of information. Uncontrollable delays introduced by the VICON server did cause a few problems in some of the experiments but did not affect significantly the performance of the system (see (Labrosse, 2006) for more details).

4.2 The datasets

The datasets cover an indoors and two outdoors environments[4]. The indoors datasets have all been grabbed in the same environment but with different trajectories and/or different objects visible in the images:

- RotateRight: a pure rotation (on the spot), a few objects surrounding the robot;
- Straight1: a pure translation, a few objects close to the robot;
- Straight2: a pure translation, no object in the vicinity of the robot (but at the extremities of the trajectory);
- Circle1: a circular trajectory, a few objects outside the trajectory, some close to it;
- Circle2: a circular trajectory, many colourful objects outside and inside the trajectory. In particular, a bright red box at the centre of the trajectory appears almost

[4]The author will gladly provide copies of the datasets if requested.

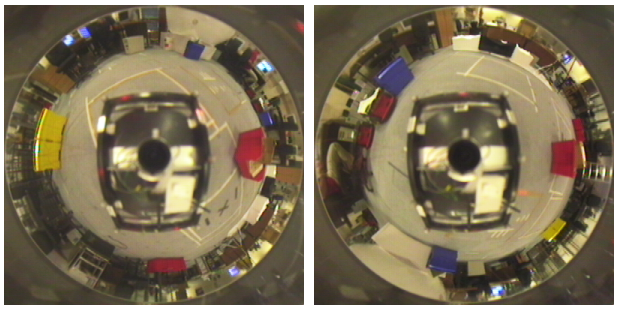

Figure 15: Two omni-directional images from CIRCLE2

static in the images (on the right of the omni-directional images of Figure 15).

Orientations provided by the visual compass are evaluated against orientations provided by the VICON system.

The ground truth for the outdoors datasets is obtained using a magnetic compass, since the VICON system is not available outdoors. The first two datasets are only used to evaluate the performance of the magnetic compass:

- PAVEMENTSTRAIGHT: a pure translation on a straight pavement between a road and a grassy area;
- GRASSSTRAIGHT: a pure translation on the grass by the pavement used in experiment PAVEMENTSTRAIGHT;
- GRASS1: a random trajectory in a grassy area surrounded by bushes;
- GRASS2: a random trajectory on a grassy area surrounded by bushes and under trees.
- CARPARK: a random trajectory on a flat but rough surface in a car park almost empty of cars;

The datasets acquired on the grassy area all contain numerous frames showing moving cars and people. The dataset of the car park contains large variations in illumination between successive images because of shadows created by the buildings around the car park.

4.3 Results

For each dataset, the orientation as a function of image index is computed using two methods. The first is the automatic algorithm described in Algorithm 12, Section 3.5, using the parameters recapitulated in Table 2. The results of this method are compared with that of a method that uses the fixed, manually determined spatial sampling (simulated by skipping a fixed number of frames) that provides the lowest error at the end of the series of images[5]. The former is referred to as "Atm" along with the threshold t used, 0.6055 being displayed as 0.6, while the latter is referred to as "Fx" along with the number of skipped frames. The results are evaluated using various measures. The maximum of the absolute value of the

[5]This is similar to the method used to determine the threshold t used in Algorithm 1, Section 3.4.4.

error (the difference between the result of the proposed method and ground truth) and the mean and standard deviation of the error are used to assess the variation of the error on each dataset. These values are obtained over whole datasets, not just considering the error at their end, unlike what was done in Section 3.4. The slope of the robust linear regression of the error as a function of time and distance travelled is used to evaluate the trend of the error. The distance travelled is not always applicable or available (the VICON system not being available outdoors and not having a GPS, outdoors experiments don't contain this information). However, all datasets have been acquired with the robot driving at similar speeds (although not exactly the same speed due, for example, to difficulties in driving on grass). This means that the slopes as a function of time are roughly comparable, although they would depend on factors such as robot speed and frame rate if these had not been controlled. When the robot does not undergo any translation (RotateRight), i.e. almost the perfect case, the error drifts linearly (Labrosse, 2006). In this case, the slope of the regression is a good measure of the drift. In more general displacements including translations, other parameters influence the performance of the system and the linear regression becomes a less appropriate model for the error. However, because of fluctuations of the error curves and because an overall measure of the drift is a good indication of the performance, linear regression has been used in all cases. The number of reference frames is given with the total number of frames in parenthesis. All the results are given in Tables 3 and 4.

Table 3 shows no surprise in the results. In all cases, the error remains low (less than 7°) and in the more general case of combined translation and rotation the error's drift also remains low (about 0.2°/m in the difficult case of Circle2 where a bright object remains static in the images — a red box at the centre of the trajectory). The manual method performs systematically slightly better than the automatic method. However, comparable results are obtained for all the datasets with parameters determined from one of these sets, the datasets being different both in the trajectory and the environment. The automatic method for Straight1 performs significantly worse than the manual method and the other datasets. This is because the environment does not vary much when seen with the reduced field of view and thus large numbers of frames are skipped, introducing significant error due to parallax. A similar example is Grass2 (below). Moreover, in the cases of Straight1 and Straight2 the rotation between frames was always negligible, thus hardly measurable, hence the larger errors.

Figures 16 and 17 show the orientation of the robot as returned by the magnetic compass and computed by the visual compass using all the images of the PavementStraight and GrassStraight datasets. The visual orientation is much smoother than the magnetic orientation. Moreover, the magnetic compass provides an orientation that is less smooth on the bumpy area of the grass than on the smooth pavement. This is because this sensor is sensitive to pitch and roll. The smoothness is obviously good for the stability of programs that might use such information to control the robot.

To capture these datasets, the robot was given constant forward speed and null rotational speed and was positioned on the straight pavement by the grassy area used in these experiments or on the grassy area by the pavement. It was thus easy to see that the actual trajectory was in fact not straight, although no actual precise measurements were taken. This was even truer for the run on grass, due to the slippery and bumpy surface. For both

Table 3: Quantitative evaluation of the performance for the different indoors datasets: statistics on the errors

Dataset	Method	Nb (total)	Max. (°)	Mean (°)	Std. dev. (°)	Slope (°/s)	Slope (°/m)
RotateRight	Fx 11	72 (853)	1.798	-0.245	0.869	0.010	n/a
	Atm 0.6	25 (853)	2.120	-0.952	0.560	-0.005	n/a
Straight1	Fx 1	118 (234)	1.737	-1.004	0.425	0.004	0.073
	Atm 0.6	15 (234)	6.479	1.910	2.047	0.102	1.694
Straight2	Fx 1	120 (238)	3.234	1.050	1.063	0.059	0.996
	Atm 0.6	13 (238)	3.992	0.971	1.540	0.068	1.133
Circle1	Fx 3	214 (851)	2.717	-0.055	1.177	-0.000	-0.003
	Atm 0.6	204 (851)	3.033	1.582	0.992	0.002	0.020
Circle2	Fx 8	94 (837)	5.001	-1.581	2.051	-0.016	-0.133
	Atm 0.6	184 (837)	6.877	-2.930	2.040	-0.025	-0.213

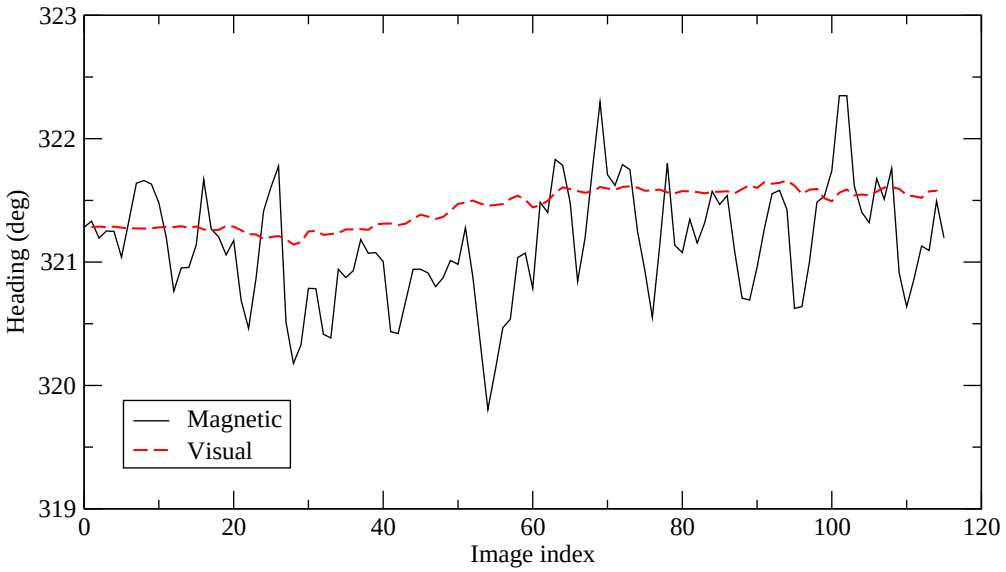

Figure 16: PavementStraight: magnetic and visual orientation with all the images

datasets, the visual orientation seems to drift. However, close inspection of the graphs shows that the magnetic orientation behaves in the same way. This shows that the apparent drift is in fact due to the small amounts of rotation the robot undertook. These experiments show that errors of about 5.5° can be due to the noisy behaviour of the magnetic compass.

It is interesting to note that frames 100 to 106 of GrassStraight contain a moving car that was large in the images and brighter than the rest of the pixels on the images. This has no effect on the visual compass because when the car was at its largest in the view, it was out of the narrowed field of view used for the orientation computation.

The experiments with the outdoors datasets show interesting problems as well as a generally good performance. In order to show the "normal" performance of the system, the statistics

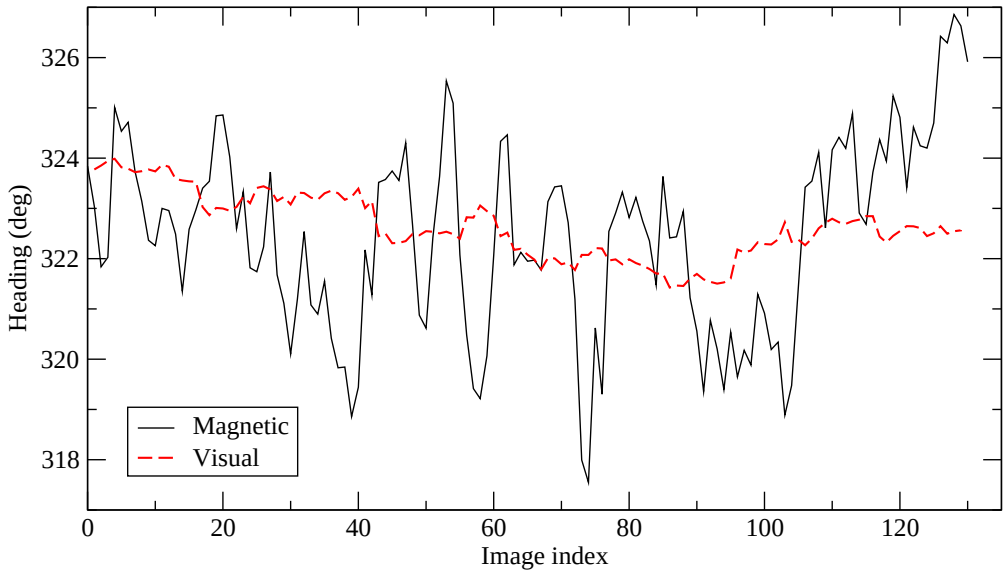

Figure 17: GrassStraight: magnetic and visual orientation with all the images

Table 4: Quantitative evaluation of the performance for the different outdoors datasets (respectively frames 0–700 and 70–1111 for starred sets): statistics on the errors

Dataset	Method	Nb (total)	Max. (°)	Mean (°)	Std. dev. (°)	Slope (°/s)
Grass1*	Fx 0	701 (701)	16.62	4.33	5.64	-0.01
	Atm 0.6	398 (701)	17.52	4.46	7.06	-0.06
Grass2*	Fx 2	349 (1043)	65.45	-15.46	20.16	-0.16
	Atm 0.6	259 (1043)	77.26	-30.74	21.69	-0.23
CarPark	Fx 3	216 (860)	36.65	2.59	13.67	0.07
	Atm 0.6	196 (860)	31.99	4.36	13.49	0.08

shown in Table 4 for the datasets Grass1 and Grass2 were calculated only on a subset of the whole dataset because of these problems.

Figure 18 shows the error between the magnetic and visual orientations for the dataset Grass1. As can be seen, up to about frame 700, the error remains below 20° but jumps to a high value in a few frames (Table 4 shows the statistics taken up to frame 700). The sudden jump is due to a combination of several effects: a not very fast rotation, a bump in the ground attenuating the effect of the rotation in the images and a bland and featureless environment, especially in the field of view kept for the computation. This will be discussed further in Section 5.

Without this effect, the maximum error is significantly worse than in the indoors cases (17° against 7°). The main differences between the indoors datasets and outdoors datasets are (1) that the outdoors datasets are visually asymmetric containing large salient features on one side of the environment and (2) that the ground is not even in the outdoors case making the match between successive images less good. This will be discussed further in Section 5.

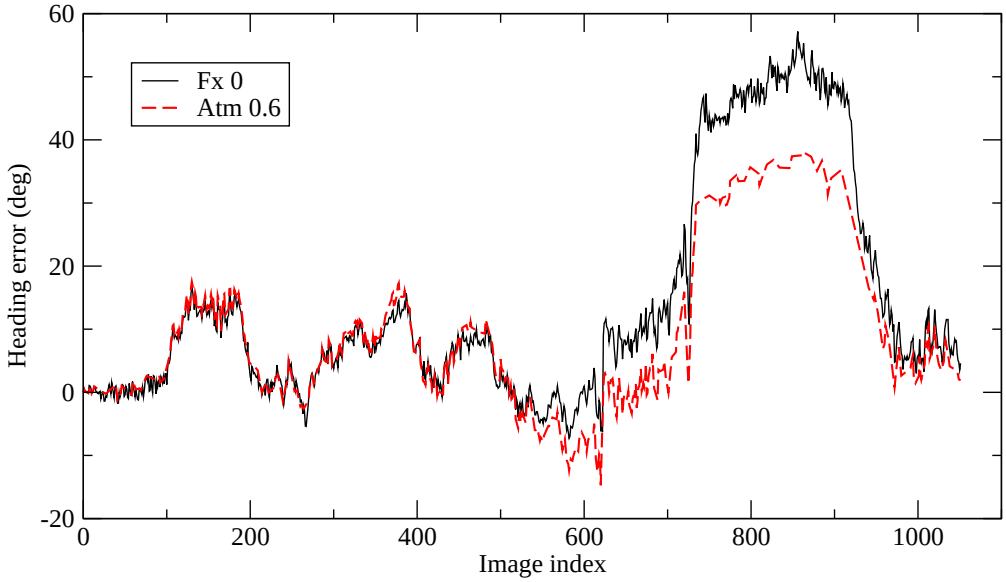

Figure 18: GRASS1: error between the magnetic and visual orientations

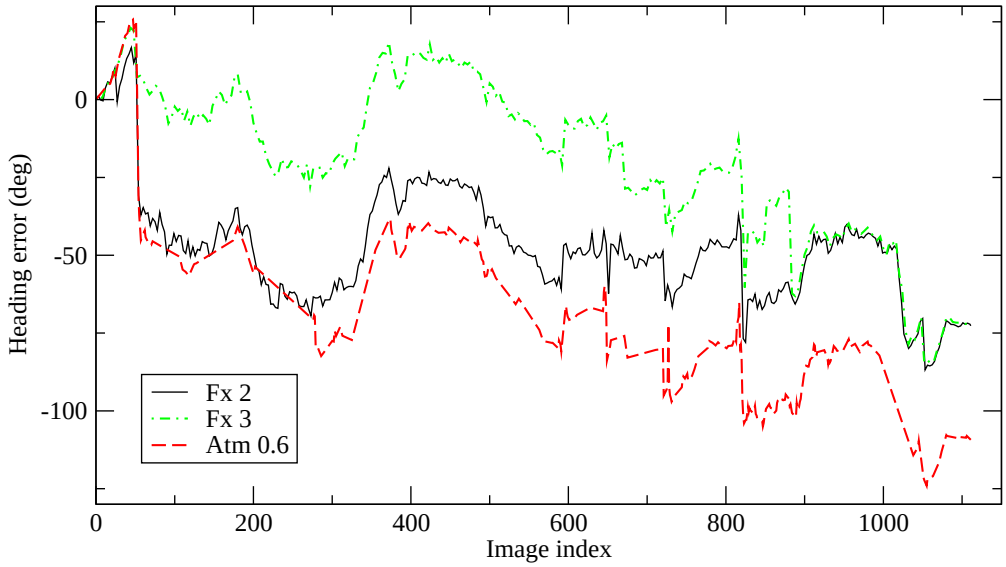

Figure 19: GRASS2: error between the magnetic and visual orientations

These two causes contribute to the sudden increases in absolute value of the error. However, this error of 17° is to be mitigated by up to 5.5° due to the noisiness of the orientation provided by the magnetic compass.

Figure 19 shows the error between magnetic and visual orientations for the dataset GRASS2. At first sight, the performance with this dataset seems much worse than with the others. This is due to a combination of problems. The first one happens between frames 51 and 54, shown in Figure 20. As in some previous cases, a bright car is passing by the robot. This time however, the car is in the kept field of view. Moreover, the other half of the field of view (front) is almost featureless, resulting in a wrong estimation of the rotation. The

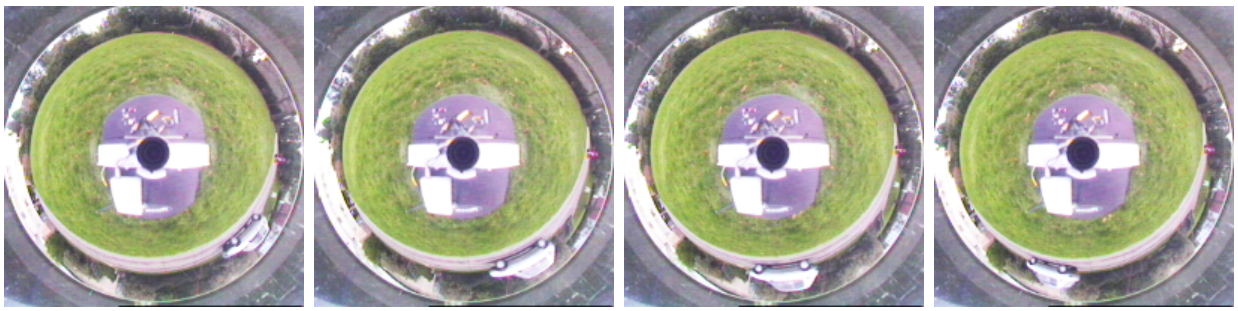

Figure 20: GRASS2: frames 51 to 54

graph obtained when skipping three frames at each measurement shows that if these frames are indeed avoided, then the problem does not happen. Figure 21 shows the distance as a function of rotation between frame 51 and frames 51 to 53, when the car was passing in the field of view. The curves are qualitatively different from the "normal" curve, e.g. the one for image 51 and itself. In particular, the global minimum of the distance does not correspond to the rotation of the robot. It does however correspond to "pushing" the car out of the field of view: Figure 22 shows the panoramic images corresponding to frames 51 to 53, frames 52 and 53 having been un-rotated according to the global minimum of the distance to frame 51. It is clear that the car is pushed out of the field of view. Indeed, the car not being visible in the kept field of view of frame 51 and because it is different from everything else in the images, the best match between frame 51 and subsequent frames corresponds to having the car out of the field of view, yet keeping the changes between the images as low as possible for the remaining parts of the images. This results in a measured rotation that corresponds to pushing the car just out of the field of view. The distance function from frame 51 to frames 52 and 53 does present two marked minima (marked 1, 2, 3 and 4 on Figure 21) that correspond to pushing the car either way out of the field of view, the higher one being for the largest image rotation.

Note that the minimisation procedure will not reach the global minimum of the distance between images 51 and 53 (marked 4 on Figure 21) but rather the local minimum close to 0° (marked 5 on Figure 21), which does correspond to the true rotation between the two frames. However, because the minimum is much higher than what it should be (because overall the second image is very different from the first because of the car), the normalised amplitude goes below the threshold and the rotation is thus calculated with the previous image (52), for which the "good" minimum is merged with the one corresponding to pushing the car out of the field of view, hence the introduced error.

Table 4 gives the statistics for GRASS2 measured from frame 70. Despite this, the error is still much larger than with the other datasets and is drifting significantly more. This happens from around frame 400, at which point the robot arrives under trees, the grass largely disappearing, thus only having brown/grey/dark green colours in its view that also became darker. All this contributes to having less contrast and thus a not very good match. Moreover, because of the tree trunks, large parallax errors were also introduce.

Finally, the CARPARK dataset shows good results, Figure 23, this time the automatic method

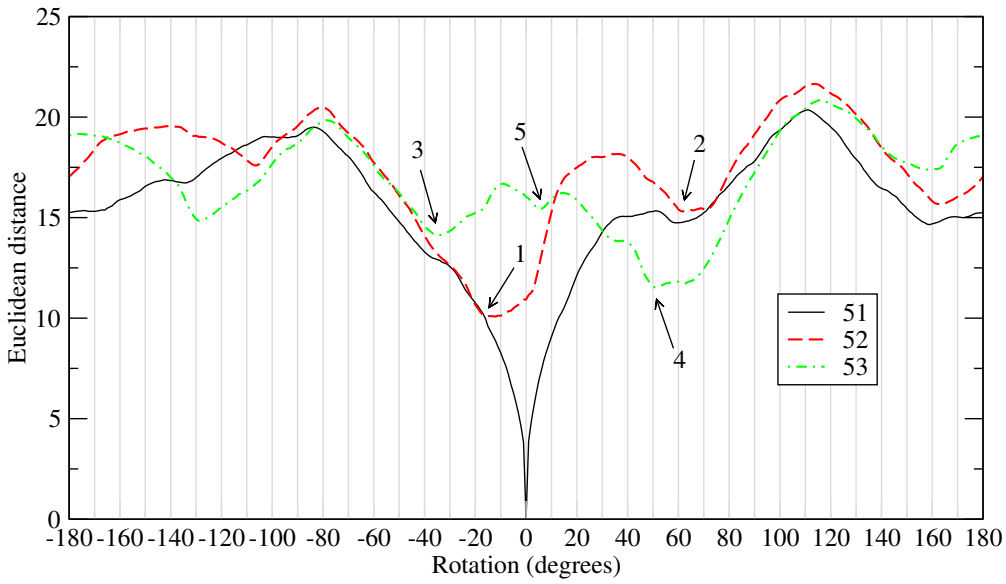

Figure 21: GRASS2: distance between image 51 and images 51 to 53 corresponding to the car passing in the field of view of the robot, Figure 20

Figure 22: GRASS2: the panoramic images corresponding to frame 51 (top) and frames 52 and 53 (middle and bottom) after un-rotation using the global minimum of the distance function in Figure 21. The light bands correspond to the kept field of view.

being slightly better than the manual one. However, important variations are visible. These are due to the fact that the environment was visually largely asymmetric (mostly due to inhomogeneous lighting and shadows) with one side much brighter than the other, Figure 24. As mentioned before, this tends to "pull" the orientation in one way or the other, depending on the rotation. However, the systems recovers because the whole of the environment was visible from everywhere and the robot was overall turning by the same amount right and left, thus cancelling out errors.

5 Discussion

We discuss here several aspects shown during the experiments reported above. Some of these aspects were expected and have been introduced in Section 3.3.5.

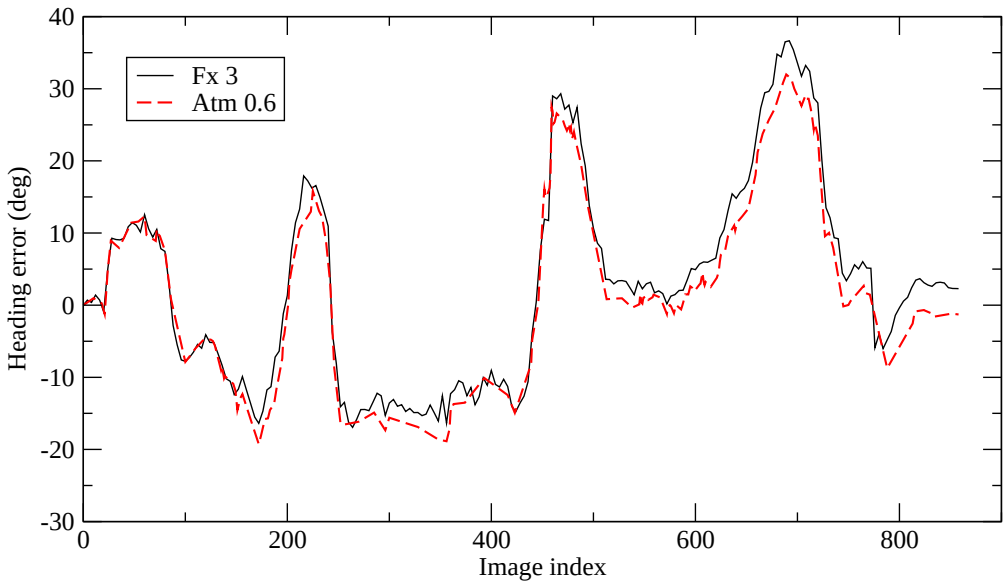

Figure 23: CarPark: error between the magnetic and visual orientations

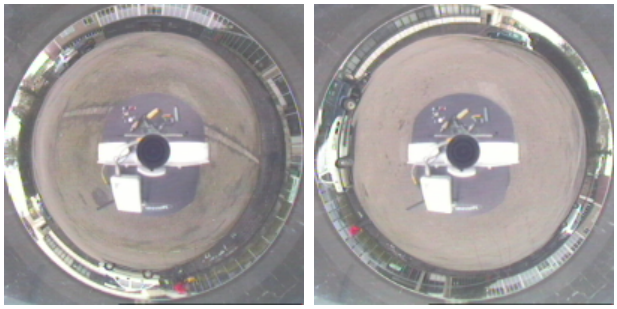

Figure 24: CarPark: frames 26 and 198

The proposed system generally copes well with dynamic environments. In all cases but one, the moving cars and people did not affect the performance of the system. This is partly due to the narrowed field of view, in effect eliminating most moving objects from the view, but also to the relative small size in the images of these objects, because of the projection on the mirror. However, reducing the field of view constrains the environment even further as this requires features everywhere (the isotropy of the environment pushed to the extreme). Moreover, if large moving objects happen to be in the field of view, then the system introduces an error related to the projection of the motion of the object. The relationship is more or less strong depending on the amount of static features present in the remaining parts of the field of view and on the visual distinctiveness of the moving object.

For example, red $(1.0, 0.0, 0.0)$ will appear much less different from black $(0.0, 0.0, 0.0)$ than purple $(1.0, 0.0, 1.0)$ from black (1 against $\sqrt{2} = 1.4142$). However, from the system's point of view, the difference purple-black should not be more important than the difference red-black. This varied influence between pairs of colours can create an inhomogeneous "pull" of the rotation towards the most distinctive colour (or indeed "push" when the colour is not visible in one frame because of the field of view but becomes visible in subsequent ones).

Note that the same would happen with different shades of grey.

The actual effect of this on the estimation of the orientation is difficult to quantify. Indeed, it depends on the distribution of colours in the images and the apparent motion in the images of the prominent colour. This apparent motion in turn depends on the motion of the robot relative to the object producing this colour and on the distance between the robot and the object.

This shows that a better combination of colour space and distance metric must be found. However, all colour spaces will use different quantifications for the different represented colours, thus producing effects similar to the one described above. Different metrics need to be evaluated. Apart from the two we mentioned before, many are available such as the traditional cross-correlation, zero-mean normalised cross-correlation and zero-mean sum of squared differences, or the less usual mutual information (Chen, 2004). These have been the object of much work in the area of image registration and will be evaluated in the future in the context of this work.

Another effect of moving objects is that successive images appear very different from each other, forcing the system to use more frames, which is generally not desirable, particularly in such situations. This could possibly be solved by running two instances of the algorithm at the same time with different values of the threshold on the distance amplitude. This would at least allow the detection of something going wrong and possibly the recombination of the two results to obtain a more robust estimation of the orientation. Another solution could be to use an idea similar to the one behind the mapping algorithm in (Neal and Labrosse, 2004): images seldom seen are at first used (in the topological map) and then discarded because not judged "strong" enough (too different from neighbouring images and thus not reinforced by them). When an image gets discarded, the orientation could be recomputed from the kept images immediately surrounding it. This needs to be tested and will be when the visual compass is integrated with the mapping algorithm (see below). An easier solution would be to use another sensor that gives a possibly rough estimation of the rotation of the robot, such as odometry or inertial sensors. This extra information could be used to detect when something goes wrong by comparing the rotation provided by both systems. In case of strong disagreement, the current frame could just be dropped to avoid accumulating wrong measurements.

The width of the front and back field of view could be adapted according to the images. For example, lack of features can be easily detected with simple methods such as colour variance in the images. Moving objects in the field of view also make the distance as a function of the rotation qualitatively different from the "normal" distance function, a fact that could be detected. When such a situation is detected, the field of view could then be increased and/or the space sampling rate modified to tackle these problems.

Despite these problems, we have shown that the algorithm described performs well, providing an estimation of the orientation with an error generally well below the maximum error insects (Cartwright and Collett, 1983; Åkesson and Wehner, 2002) and algorithms derived from the snapshot model can cope with ((Ruchti, 2000) gives a maximum error of 45°). Perhaps more importantly, results show that the error drifts only very slowly, which makes it suitable for

tasks such as homing or even long-range navigation where the path is specified in terms of visual targets. Moreover, it has been shown that the performance of the system can be improved even further by using higher resolution images. Although this does imply heavier processing, the computations performed in the system being only low level ones, they could be easily implemented in hardware and would thus be faster, and not using processing power of the computer, allowing the use of higher resolution images.

Inertial navigation systems are nowadays often used in robotic applications to incrementally compute the pose of the robot while it moves. Studies on the performance of such systems to estimate the orientation reveal that drift of error can be as high as 1.35°/s (Barshan and Durrant-Whyte, 1995), an order of magnitude higher than our system. A better performance can be obtained by inertial sensors when their output is used in an (extended) Kalman filter. In that case, drifts of the order of 3°/min can be obtained but can still produce errors in the orientation estimation as high as 12° (Barshan and Durrant-Whyte, 1995) or more recently around 9° (Hogg et al., 2002). This shows that our visual compass performs better than inertial sensors and could be improved even further by integrating it with or without other sensors in a Kalman filter (or more generally an extended information filter).

Finally, we have seen that our system provides much smoother orientations than the one provided by the magnetic compass used in these experiments. Moreover, our system is not too sensitive to pitch and roll. The only constraint is that images contain features that are tall enough to overlap between successive images in case of pitch and roll. This is often the case as many environments contain tall objects (buildings, trees, mountains, etc.).

However, if the terrain becomes rough enough images could change dramatically, which would deteriorate significantly the performance of our system. We are currently envisaging the use of passively or actively controlled stabilisation platforms for the camera.

The visual compass is successfully used in an appearance-based implementation of visual homing to provide orientation information needed by the method (Binding and Labrosse, 2006). We also intend on integrating the visual compass in a mapping algorithm using similar techniques, early versions of which having been presented in the past (Neal and Labrosse, 2004), in particular to help disambiguate image matches. These will allow frequent re-localisation and thus re-alignment of the orientation, similar to procedures insects seem to adopt (Srinivasan et al., 1997).

6 Conclusion

We have introduced a method to incrementally estimate the orientation of a robot using sub-symbolic matching of successive panoramic images grabbed from the robot. We have seen that many factors can contribute positively and/or negatively to the performance of the method and we have performed a careful study with real data of some of these factors. Others remain to be studied, such as the combination of colour space and distance metric. Angular resolution of the panoramic images has been shown to be very effective at improving the performance of the system but has not been used because computationally expensive. However, interpolation of the distance function has been shown to be not as efficient but at

a fraction of the cost and has thus been retained. Reducing the field of view to the front and back of the robot reduces the computational cost and improves the robustness of the system because this eliminates most of the translational optic flow and eventual objects moving on the side of the robot. Finally, the effect of spatial sampling has been characterised and a method to estimate when successive images are becoming too different has been proposed, allowing a good trade-off between integrating many small errors or not many larger errors.

An algorithm implementing a solution to the problem has been presented and results using real data acquired in un-modified indoors and outdoors dynamic environments have been presented. Problems inherent more to the visual aspect of the method rather than to the algorithm itself have been discussed. In particular, we have given evidence that the combination of RGB colour space and Euclidean distance, although provides good results most of the times, can be not good enough in some situations, for example when the appearances present colours that are stronger in value than others and are not uniformly distributed. Apart from increasing the angular resolution of the panoramic images (only difficult due to the extra computation involved), this is probably where the method can be improved most. We also have proposed a method to relate apparent changes in the environment to changes in the image space.

Despite these less than optimal choices in distance metric and colour space and with the low angular resolution of the panoramic images, the performance of the system has been shown to be good and drifting only slowly, which is an important characteristic, especially since the algorithm proceeds by integrating local changes in orientation. In fact, the performance of our system is similar to that of traditional inertial sensors, if not better. Moreover it uses the same techniques as the ones used in the overall SLAM system we are currently working on, therefore sharing some of the computation and not introducing more complexity and the need for other sensors.

Acknowledgement

The equipment used for this work was partly funded by HEFCW Science Research Investment Fund (SRIF) grants from 1999 (for the VICON system) and 2002 (for the research lab and the robotic equipment).

The author gratefully acknowledges the help from Dr Mark Neal provided during many animated discussions as well as the comments from the reviewers that did help improve the paper.

References

Åkesson, S. and Wehner, R. (2002). Visual navigation in desert ants *Cataglyphis fortis*: are snapshots coupled to a celestial system of reference? *Journal of Experimental Biology*, 205:1971–1978.

Barshan, B. and Durrant-Whyte, H. F. (1995). Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation*, 11(3):328–342.

Bichsel, M. and Pentland, A. P. (1994). Human face recognition and the face image set's topology. *CVGIP: Image Understanding*, 59(2):254–261.

Binding, D. and Labrosse, F. (2006). Visual local navigation using warped panoramic images. In *Proceedings of Towards Autonomous Robotic Systems*, pages 19–26, University of Surrey, Guildford, UK.

Bisset, D. L., Aldred, M. D., and Wiseman, S. J. (2003). Light detection apparatus. United States Patent US 6,590,222 B1. Also UK Patent GB 2 344 884 A, 2000.

Cartwright, B. A. and Collett, T. S. (1983). Landmark learning in bees: experiments and models. *Journal of Comparative Physiology*, 151:521–543.

Cartwright, B. A. and Collett, T. S. (1987). Landmark maps for honeybees. *Biological Cybernetics*, 57(1/2):85–93.

Chen, H.-m. (2004). Mutual information: A similarity measure for intensity based image registration. In Varshney, P. K. and Arora, M. K., editors, *Advanced Image Processing Techniques for Remotely Sensed Hyperspectral Data*, chapter 3, pages 89–108. Kluwer.

Cozman, F. and Krotkov, E. (1995). Robot localization using a computer vision sextant. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 106–111.

Cozman, F., Krotkov, E., and Guestrin, C. (2000). Outdoor visual position estimation for planetary rovers. *Autonomous Robots*, 9(2):135–150.

Franz, M. O., Schölkopf, B., and Bülthoff, H. H. (1997). Homing by parameterized scene matching. In *Advances in Artificial Life: Proceedings of the European Conference on Artificial Life*, pages 236–245.

Franz, M. O., Schölkopf, B., Mallot, H. A., and Bülthoff, H. H. (1998). Where did I take that snapshot? Scene-based homing by image matching. *Biological Cybernetics*, 79:191–202.

Frier, H. J., Edwards, E., Smith, C., Neale, S., and Collett, T. S. (1996). Magnetic compass cues and visual pattern learning in honeybees. *Journal of Experimental Biology*, 199(6):1353–1361.

Gaspar, J., Winters, N., and Santos-Victor, J. (2000). Vision-based navigation and environmental representations with an omnidirectional camera. *IEEE Transactions on Robotics and Automation*, 16(6):890–898.

Goedemé, T., Tuytelaars, T., Van Gool, L., Vanhooydonck, D., Demeester, E., and Nuttin, M. (2005). Is structure needed for omnidirectional visual homing? In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 303–308.

Gonzales-Barbosa, J.-J. and Lacroix, S. (2002). Rover localization in natural environments by indexing panoramic images. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1365–1370, Washington, USA.

Gourichon, S. (2004). *Utilisation d'un compas visuel pour la navigation d'un robot mobile*. PhD thesis, Université Paris VI, Paris, France.

Graham, P., Fauria, K., and Collett, T. S. (2003). The influence of beacon-aiming on the routes of wood ants. *Journal of Experimental Biology*, 206:535–541.

Hogg, R. W., Rankin, A. L., Roumeliotis, S. I., McHenry, M. C., Helmick, D. M., Bergh, C. F., and Matthies, L. (2002). Algorithms and sensors for small robot path following. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3850–3857, Washington DC, USA.

Jogan, M. and Leonardis, A. (2000a). Robust localization using eigenspace of spinning-images. In *Proceeding of the IEEE Workshop on Omnidirectional Vision*, pages 37–46, Hilton Head Island, South Carolina, USA.

Jogan, M. and Leonardis, A. (2000b). Robust localization using panoramic view-based recognition. In *Proceedings of the International Conference on Pattern Recognition*, volume 4, pages 136–139, Barcelona, Spain.

Labrosse, F. (2004). Visual compass. In *Proceedings of Towards Autonomous Robotic Systems*, pages 85–92, University of Essex, Colchester, UK.

Labrosse, F. (2006). Appearance-based heading estimation: the visual compass. Technical Report UWA-DCS-06-048, Computer Science Department, University of Wales, Aberystwyth, UK.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110.

Makadia, A. and Daniilidis, K. (2006). Rotation recovery from spherical images without correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1170–1175.

Mitchell, T. and Labrosse, F. (2004). Visual homing: a purely appearance-based approach. In *Proceedings of Towards Autonomous Robotic Systems*, pages 101–108, University of Essex, Colchester, UK.

Möller, R. (2001). Do insects use templates or parameters for landmark navigation? *Journal of Theoretical Biology*, 210(1):33–45.

Möller, R., Maris, M., and Lambrinos, D. (1999). A neural model of landmark navigation in insects. *Neurocomputing*, 26–27:801–808.

Naval Jr., P. C., Mukunoki, M., Minoh, M., and Ikeda, K. (1997). Estimating camera position and orientation from geographical map and mountain image. In *Proceedings of the 38th Research Meeting of the Pattern Sensing Group, Society of Instrument and Control Engineers*, pages 9–16, Tokyo, Japan.

Nayar, S. K., Nene, S. A., and Murase, H. (1996). Subspace methods for robot vision. *IEEE Transactions on Robotics and Automation*, 12(5):750–758.

Neal, M. and Labrosse, F. (2004). Rotation-invariant appearance based maps for robot navigation using an artificial immune network algorithm. In *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 863–870, Portland, Oregon, USA.

Nelson, R. C. and Aloimonos, J. (1988). Finding motion parameters from spherical flow fields (or the advantages of having eyes in the back of your head). *Biological Cybernetics*, 58:261–273.

Nistér, D., Naroditsky, O., and Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1).

Pajdla, T. and Hlaváč, V. (1999). Zero phase representation of panoramic images for image based localization. In *Proceedings of Computer Analysis of Images and Patterns*, pages 550–557, Ljubljana, Slovenia.

Röfer, T. (1995). Image based homing using a self-organizing feature map. In *Proceeding of the International Conference on Artificial Neural Networks*, volume 1, pages 475–480.

Röfer, T. (1997). Controlling a wheelchair with image-based homing. In *Proceedings of the AISB Symposium on Spatial Reasoning in Mobile Robots and Animals*, Manchester University, UK.

Ruchti, S. (2000). Landmark and compass reference in landmark navigation. Master's thesis, Artificial Intelligence Lab des Institutes für Informatik der Universität Zürich.

Se, S., Lowe, D., and Little, J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758.

Shaw, A. and Barnes, D. (2003). Landmark recognition for localisation and navigation of aerial vehicles. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 1, pages 42–47.

Srinivasan, M. V., Zhang, S. W., and Bidwell, N. J. (1997). Visually mediated odometry in honeybees. *Journal of Experimental Biology*, 200:2513–2522.

Thompson, W. B., Henderson, T. C., Colvin, T. L., Dick, L. B., and Valiquette, C. M. (1993). Vision-based localization. In *Proceedings of DARPA Image Understanding Workshop*, pages 491–498.

Vardy, A. and Oppacher, F. (2003). Low-level visual homing. In *Advances in Artificial Life: Proceedings of the European Conference on Artificial Life*.

Vassallo, R. F., Santos-Victor, J., and Schneebeli, H. J. (2002). Using motor representations for topological mapping and navigation. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 478–483, Lausanne, Switzerland.

Wehner, R., Michel, B., and Antonsen, P. (1996). Visual navigation in insects: coupling egocentric and geocentric information. *Journal of Experimental Biology*, 199:129–140.

Woodland, A. and Labrosse, F. (2005). On the separation of luminance from colour in images. In *Proceedings of the International Conference on Vision, Video, and Graphics*, pages 29–36, University of Edinburgh, UK.

Zeil, J., Hofmann, M. I., and Chahl, J. S. (2003). Catchment areas of panoramic snapshots in outdoor scenes. *Journal of the Optical Society of America A*, 20(3):450–469.

Zeil, J., Kelber, A., and Voss, R. (1996). Structure and function of learning flights in bees and wasps. *Journal of Experimental Biology*, 199:245–252.